Monte Carlo mini-App — Exa.py The Python Framework

When developing implementations of "mini-apps" or "co-design applications" for Exascale R&D, it is important to demonstrate an algorithm's potential in several different forms, perhaps including:

- Serial implementations
- Standard parallel implementations (MPI, OpenMP)
- Emerging parallelism frameworks (CUDA, OpenCL)

The goal of the ***Exa*** Python framework [3] is to provide an easy way to implement an algorithm once and run it, near-optimally, on multiple different forms of potential Exascale technologies.

Two basic concepts are utilized in this framework, enabling algorithms written with ***Exa*** to be run on multiple platforms with one source:

–Array operations: Unary or Binary

- Operations on each element of an array or arrays resulting in a new array

–Array reductions: Reduction to a single element

- Summations, minimums, maximums…

The framework currently has a defined methodology hierarchy:

1. CUDA – if available
2. OpenCL – if available
3. MPI – if available
4. OpenMP – if available
5. Serial execution

The user can also choose the method to use for each algorithmic element.

Larry Cox, ljcox@lanl.gov, (505) 665-7344

Los Alamos NATIONAL LABORATORY EST. 1943

NNSA National Nuclear Security Administration